

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-240704

(43)Date of publication of application : 11.09.1998

(51)Int.Cl. G06F 15/16
G06F 15/16

(21)Application number : 09-065468 (71)Applicant : RICOH CO LTD
(22)Date of filing : 04.03.1997 (72)Inventor : SATO MASAKI
SUDO YASUSHI

(30)Priority

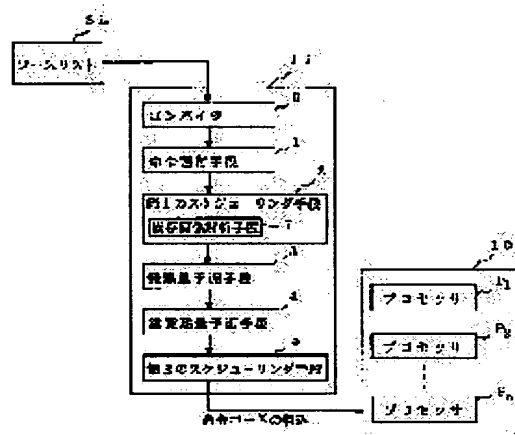
Priority number : 08356843 Priority date : 26.12.1996 Priority country : JP

(54) MULTIPROCESSOR SYSTEM AND INSTRUCTION GENERATING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To perform processing without exceeding the heat generation permissible limit of a chip by rescheduling instructions by processors on a chip according to the predicted total heating value, and suppressing the heat generation of the whole chip.

SOLUTION: When allocating instructions which are compiled to respective processors P1 to Pn, a 1st scheduling means 2 schedules execution order, etc., by the instructions allocated to the processors P1 to Pn, and a heating value predicting means 3 predicts the heating value of a single processor body by the instructions. Thus, a total heating value predicting means 4 predicts the heating value of the whole chip 10 from the heating values of the respective processors P1 to Pn and a 2nd scheduling means 5 reschedules the instructions by the processors P1 to Pn scheduled by the 1st scheduling means 2 according to the predicted total heating value, thereby suppressing the heating value of the whole chip 10.



LEGAL STATUS

[Date of request for examination] 18.02.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3567354

[Date of registration]

25.06.2004

BEST AVAILABLE COF

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-240704

(43) 公開日 平成10年(1998) 9月11日

(51) Int.Cl.⁸
G 0 6 F 15/16識別記号
4 5 0
4 3 0F I
G 0 6 F 15/164 5 0 Z
4 3 0 A

審査請求 未請求 請求項の数 7 F D (全 11 頁)

(21) 出願番号 特願平9-65468

(22) 出願日 平成9年(1997) 3月4日

(31) 優先権主張番号 特願平8-356843

(32) 優先日 平8(1996)12月26日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 佐藤 正喜

東京都大田区中馬込1丁目3番6号 株式
会社リコー内

(72) 発明者 須藤 靖

東京都大田区中馬込1丁目3番6号 株式
会社リコー内

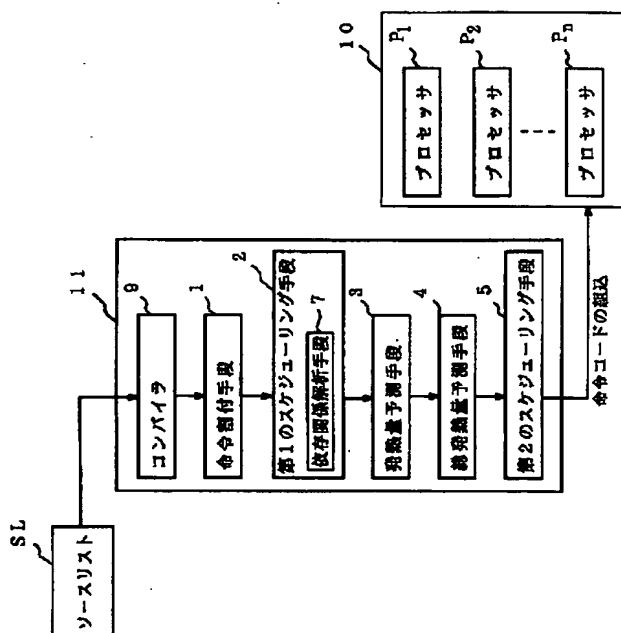
(74) 代理人 弁理士 植本 雅治

(54) 【発明の名称】 マルチプロセッサシステムおよび命令作成装置

(57) 【要約】

【課題】 複数のプロセッサが実装されたチップ全体の発熱量を抑える。

【解決手段】 命令作成装置11は、ソースリストSLをコンパイルするコンパイラ9と、コンパイル後の命令をチップ10上の各プロセッサP₁～P_nへ割付ける命令割付手段1と、各プロセッサP₁～P_n毎に割付けられた命令間の同期をとるように命令のスケジューリングを行なう第1のスケジューリング手段2と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段3と、各プロセッサP₁～P_nの発熱量からチップ10全体の発熱量を予測する総発熱量予測手段4と、総発熱量予測手段4によって予測された総発熱に基づいて、第1のスケジューリング手段2によってスケジューリングされた各プロセッサ毎の命令の再スケジューリングを行ないチップ10全体の発熱量を抑制する第2のスケジューリング手段5とを有している。



【特許請求の範囲】

【請求項1】 独立動作可能な複数のプロセッサがチップ上に実装されているマルチプロセッサシステムにおいて、コンパイル後の命令をチップ上の各プロセッサに割り付ける命令割付手段と、各プロセッサ毎に割り付けられた命令間のデータ依存関係、制御依存関係を解析し、同期をとるように命令のスケジューリングを行なう第1のスケジューリング手段と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段と、各プロセッサの発熱量からチップ全体の発熱量を予測する総発熱量予測手段と、総発熱量予測手段によって予測された総発熱量からチップ上の各プロセッサ毎の命令の再スケジューリングを行ないチップ全体の発熱量を抑制する第2のスケジューリング手段とを有していることを特徴とするマルチプロセッサシステム。

【請求項2】 請求項1記載のマルチプロセッサシステムにおいて、前記第1のスケジューリング手段は、実行待ち状態にある複数の命令中から次の実行の対象となる命令候補を選択するようになっており、前記発熱量予測手段は、第1のスケジューリング手段によって選択された命令候補についてプロセッサ単体の命令ステップ単位の発熱量の予測値を算出し、また、前記総発熱量予測手段は、各プロセッサの命令ステップ単位の発熱量の予測値から命令ステップ単位でのチップ全体の発熱量を予測し、さらに、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算し、該単位時間の間の累積加算値を総発熱量として算出し、前記第2のスケジューリング手段は、総発熱量予測手段により算出された総発熱量が制限値を越える場合には、この制限値を越えないように、第1のスケジューリング手段によってスケジューリングされた各プロセッサ毎の命令の再スケジューリングを行なうようになっていることを特徴とするマルチプロセッサシステム。

【請求項3】 請求項1または請求項2記載のマルチプロセッサシステムにおいて、前記第2のスケジューリング手段から前記第1のスケジューリング手段へのフィードバックを行ない、前記第2のスケジューリング手段によって再スケジューリングされた命令を前記第1のスケジューリング手段によって再びスケジューリングする処理を、所定回数、繰り返し行なうことを特徴とするマルチプロセッサシステム。

【請求項4】 独立動作可能な複数のプロセッサがチップ上に実装されているマルチプロセッサシステムにおいて、チップ上の各プロセッサが実行する命令を作成する命令作成装置であって、該命令作成装置は、コンパイル後の命令をチップ上の各プロセッサに割り付ける命令割付手段と、各プロセッサ毎に割り付けられた命令間のデータの同期をとるように命令のスケジューリングを行なう第1のスケジューリング手段と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段と、各

プロセッサの発熱量からチップ全体の発熱量を予測する総発熱量予測手段と、総発熱量予測手段によって予測された総発熱量からチップ上の各プロセッサ毎の命令の再スケジューリングを行ないチップ全体の発熱量を抑制する第2のスケジューリング手段とを有していることを特徴とする命令作成装置。

【請求項5】 請求項4記載の命令作成装置において、前記第1のスケジューリング手段は、実行待ち状態にある複数の命令中から次の実行の対象となる命令候補を選択するようになっており、前記発熱量予測手段は、第1のスケジューリング手段によって選択された命令候補についてプロセッサ単体の命令ステップ単位の発熱量の予測値を算出し、また、前記総発熱量予測手段は、各プロセッサの命令ステップ単位の発熱量の予測値から命令ステップ単位でのチップ全体の発熱量を予測し、さらに、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算し、該単位時間の間の累積加算値を総発熱量として算出し、前記第2のスケジューリング手段は、総発熱量予測手段により算出された総発熱量が制限値を越える場合には、この制限値を越えないように、第1のスケジューリング手段によってスケジューリングされた各プロセッサ毎の命令の再スケジューリングを行なうようになっていることを特徴とする命令作成装置。

【請求項6】 請求項4または請求項5記載の命令作成装置において、前記総発熱量予測手段は、前記単位時間当たりの総発熱量の算出において、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算するとともに、単位時間経過した命令の発熱量を前記累積加算値から減算して、総発熱量を算出するようになっていることを特徴とする命令作成装置。

【請求項7】 請求項4または請求項5記載の命令作成装置において、前記第2のスケジューリング手段から前記第1のスケジューリング手段へのフィードバックを行ない、前記第2のスケジューリング手段によって再スケジューリングされた命令を前記第1のスケジューリング手段によって再びスケジューリングする処理を、所定回数、繰り返し行なうことを特徴とする命令作成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、独立動作可能な複数のプロセッサがチップ上に実装されているマルチプロセッサシステムおよび命令作成装置に関する。

【0002】

【従来の技術】近年、LSI製造技術の進歩に伴い、超微細化、高集積化が進み、1チップ上に複数のプロセッサを配置(実装)することが可能となってきている。そこで、チップの性能限界として新たにチップ内の発熱量の問題が考えられるようになってきた。これはチップ内の

活性化率が上がると発熱量が上昇しチップの発熱許容値を越えるために、発熱によってチップの動作限界が規定される(すなわち、処理を効率的に行なうことができない)ということをも意味する。

【0003】

【発明が解決しようとする課題】このように従来のマルチプロセッサシステムでは、チップ内の活性化率が上がると発熱量が上昇し、チップの発熱許容値を越えてしまい、効率的に処理を実行することができないという事態が生じてしまうという問題があった。

【0004】本発明は、チップ上に複数のプロセッサが実装される場合に、チップ全体の発熱量を抑えるように各プロセッサを制御し、チップ全体の発熱量を抑えつつ効率的に処理を実行することの可能なマルチプロセッサシステムおよび命令作成装置を提供することを目的としている。

【0005】

【課題を解決するための手段】上記目的を達成するために、請求項1、請求項4記載の発明では、独立動作可能な複数のプロセッサがチップ上に実装されているマルチプロセッサシステムにおいて、コンパイル後の命令を各プロセッサに割り付ける命令割付手段と、各プロセッサ毎に割り付けられた命令間の同期をとるように命令のスケジューリングを行なう第1のスケジューリング手段と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段と、各プロセッサの発熱量からチップ全体の発熱量を予測する総発熱量予測手段と、総発熱量予測手段によって予測された総発熱量からチップ上の各プロセッサ毎の命令の再スケジューリングを行ないチップ全体の発熱量を抑制する第2のスケジューリング手段とを有していることを特徴としている。

【0006】また、請求項2、請求項5記載の発明では、前記第1のスケジューリング手段は、実行待ち状態にある複数の命令中から次の実行の対象となる命令候補を選択するようになっており、前記発熱量予測手段は、第1のスケジューリング手段によって選択された命令候補についてプロセッサ単体の命令ステップ単位の発熱量の予測値を算出し、また、前記総発熱量予測手段は、各プロセッサの命令ステップ単位の発熱量の予測値から命令ステップ単位でのチップ全体の発熱量を予測し、さらに、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算し、該単位時間の間の累積加算値を総発熱量として算出し、前記第2のスケジューリング手段は、総発熱量予測手段により算出された総発熱量が制限値を越える場合には、この制限値を越えないように、第1のスケジューリング手段によってスケジューリングされた各プロセッサ毎の命令の再スケジューリングを行なうようになっていることを特徴としている。

【0007】また、請求項3、請求項7記載の発明で

は、前記第2のスケジューリング手段から前記第1のスケジューリング手段へのフィードバックを行ない、前記第2のスケジューリング手段によって再スケジューリングされた命令を前記第1のスケジューリング手段によって再びスケジューリングする処理を、所定回数、繰り返して行なうことを特徴としている。

【0008】また、請求項6記載の発明では、前記総発熱量予測手段は、前記単位時間当たりの総発熱量の算出において、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算するとともに、単位時間経過した命令の発熱量を前記累積加算値から減算して、総発熱量を算出するようになっていることを特徴としている。

【0009】

【発明の実施の形態】以下、本発明の実施形態を図面に基づいて説明する。本発明は、複数のプロセッサが実装されているマルチプロセッサシステムにおいて、並列処理を行なう場合にプロセッサ間の依存関係の問題等のために(プロセッサ間の同期をとる必要があるために)、全てのプロセッサが常に動作しているような状況はありえず(全てのプロセッサが同時に命令を実行しているような状況はありえず)、複数のプロセッサのうち、必ず、待ち状態になる(データ待ち等による処理待ち状態の)プロセッサが必ず存在することに着目してなされたものである。すなわち、本発明は、この処理待ちのプロセッサを効率的に利用するとともに処理への影響を最小限にとどめながらプロセッサの制御を行ない、発熱量を抑えることでチップ全体として発熱量を制御し、効率よく処理を実行することを意図している。

【0010】図1は本発明に係るマルチプロセッサシステムの構成例を示す図である。図1を参照すると、このマルチプロセッサシステムは、独立動作可能な複数のプロセッサ $P_1 \sim P_n$ が実装されている1つのチップ10と、ソースリストSLに基づいてチップ10上の各プロセッサ $P_1 \sim P_n$ に命令コードを割り付け、組み込むための命令作成装置11とを有している。

【0011】ここで、命令作成装置11は、例えばワークステーションなどで実現され、このマルチプロセッサシステムの開発機器として機能するようになっており、ソースリストSLをコンパイルするコンパイラ9と、コンパイル後の命令(オブジェクトコード(機械語命令))をチップ10上の各プロセッサ $P_1 \sim P_n$ に割り付ける命令割付手段1と、各プロセッサ $P_1 \sim P_n$ 毎に割り付けられた命令間の同期をとるように命令のスケジューリングを行なう第1のスケジューリング手段2と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段3と、各プロセッサ $P_1 \sim P_n$ の発熱量からチップ10全体の発熱量を予測する総発熱量予測手段4と、総発熱量予測手段4によって予測された総発熱量に基づいて、第1のスケジューリング手段2によってスケジューリング

された各プロセッサ毎の命令の再スケジューリングを行ないチップ10全体の発熱量を抑制する第2のスケジューリング手段5とを有している。

【0012】ここで、第1のスケジューリング手段2は、コンパイル後の命令を各プロセッサに割り付ける際、各プロセッサに割り付けた命令毎に実行順序等のスケジューリングを行ない、発熱量予測手段3において処理を行なう命令候補を順次選択するようになっている。

【0013】具体的には、第1のスケジューリング手段2は、各プロセッサに割り付けられた命令間のデータ依存関係、制御依存関係を解析する依存関係解析手段7を具備し、依存関係解析手段7の解析結果に基づいて、命令の実行順序の並べ替えやウェイト命令の挿入などを行なうようになっている。

【0014】また、発熱量予測手段3は、第1のスケジューリング手段2によって選択された命令候補についてプロセッサ単体の命令ステップ単位の発熱量(プロセッサ上で命令候補を実行した場合のそれぞれのプロセッサの発熱量)の予測値を算出するようになっている。

【0015】具体的に、発熱量予測手段3は、プロセッサの命令群について、予め1命令に対応した1プロセッサ分の発熱量の予測値を例えばメモリにテーブルとして記憶しておき、コンパイル後の命令(オブジェクトコード)が各プロセッサに割り付けられる時に、命令ステップ毎にこれらの値を読み込み、第1のスケジューリング手段2によって選択された命令候補についてプロセッサ単体の命令ステップ単位の発熱量の予測値を得ようになっている。なお、発熱量の予測値はシミュレーションや実験等により求めておくことができる。

【0016】また、総発熱量予測手段4は、各プロセッサの命令ステップ単位の発熱量の予測値から命令ステップ単位でのチップ全体の発熱量を予測し、さらに、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算し、該単位時間の間の累積加算値を総発熱量として算出するようになっている。

【0017】すなわち、総発熱量予測手段4は、まず、各プロセッサ $P_1 \sim P_n$ について命令ステップ毎に発熱量を求めておき、これら N 個の発熱量の総和を求めることによりチップ10全体の発熱量を得ようになっている。これが命令実行時間当たりの総発熱量となり、さらに、この値を単位時間の間だけ累積加算することにより、単位時間当たりの総発熱量を得ようになっている。

【0018】図2には、 N 個のプロセッサ $P_1 \sim P_n$ のうち、2つのプロセッサ P_1 、 P_2 だけが稼働している(命令を実行している)とした場合の発熱量予測手段3、総発熱量予測手段4の処理概要が示されている。図2を参照すると、チップ内部のプロセッサ P_1 とプロセッサ P_2 が命令を実行しているとする場合、まず、発熱量予測手

段3は、プロセッサ P_1 の実行命令の命令コード C_1 により予測値テーブルから C_1 の発熱量 H_1 を読み出し、また、プロセッサ P_2 の実行命令の命令コード C_2 により発熱量 H_2 を読み出して、プロセッサ単体の命令ステップ単位の発熱量の予測値を算出する。

【0019】次いで、総発熱量予測手段4は、これら発熱量 H_1 と H_2 の総和 S_1 を求めることにより、その命令ステップでのチップ全体の発熱量 S_1 (命令実行時間当たりの総発熱量)を得て、さらに、この発熱量 S_1 を単位時間の間、累積加算することにより、単位時間当たりの総発熱量 S_2 を得ようになっている。

【0020】なお、ここで、単位時間とは、チップ内部で発生した熱が外部に放射されるまでの時間を意味している。実際には、チップを覆うパッケージ等の熱伝達特性により影響されるため、シミュレーションや実験等から単位時間としての適正値を得ることができる。

【0021】また、第2のスケジューリング手段5は、総発熱量予測手段4により算出された総発熱量が制限値を越える場合には、この制限値を越えないように、第1のスケジューリング手段2によって選択された次の命令候補を他の命令の実行に置き換えるか、あるいは命令実行順序の入れ替えを行なうか、あるいは発熱を抑制するウェイト命令を挿入するように、第1のスケジューリング手段2によってスケジューリングされた各プロセッサごとの命令の再スケジューリングを行なうようになっている。

【0022】換言すれば、命令スケジューリングに着目するとき、他プロセッサと同期をとる部分については第1のスケジューリング手段2で行なわれ、発熱量を制限する部分については第2のスケジューリング手段5によって行なわれる。すなわち、図1のシステムでは、第1のスケジューリング手段2にて、他のプロセッサとの同期をとるように命令スケジューリングを行ない、この同期をとるための命令部分を固定した後、さらに第2のスケジューリング手段5にて、所定の命令スケジュール区間における発熱量が制限内になるように再スケジューリングを行なうようになっている。

【0023】図3は発熱量予測手段3、総発熱量予測手段4で図2のような処理がなされたとした場合の第2のスケジューリング手段5における処理の概要を説明するための図である。なお、図3の例では、説明の便宜上、1命令の実行時間間隔を1命令ステップとし、従って、図3の横軸の1つのブロック(1命令の実行時間)が1命令ステップとなっている。

【0024】また、図3において、単位時間が設定されるエリアは、前述のように、累積加算値 S_2 (対象エリア内に記述された命令に対応する発熱量の予測値(テーブル読み出し)の累積加算値)を求めるときの対象となるエリアであり、第2のスケジューリング手段5は、上記エリア(単位時間)における累積加算値 S_2 を対象エリア

内の命令数で割り、平均した結果が制限値の範囲内に収まるように命令のスケジューリングを行なう。なお、図3の例では、8命令ステップが単位時間となっている。

【0025】図4には、1つのプロセッサ、例えばP_nに対する命令スケジューリングの一例が示されている。図4の例では、このプロセッサP_nに割り当てられた命令スケジューリングは、K個の命令スケジューリング区間A₁～A_kとして求められる。この場合、K個の命令スケジューリング区間A₁～A_kのうちの1つの区間が上記単位時間(発熱量算出範囲)に相当したものとなっている。換言すれば、図4の例では、命令スケジューリング区間は、単位時間と同じ命令ステップ数からなっている。

【0026】このように、この第1の構成例では、この対象エリアを区間分割することにより、区間毎に累積加算値S2を得る。このため、S2はその区間に記述された命令に対応する発熱量の予測値を全て累積加算することにより得られる。図3を参照すると、第2のスケジューリング手段5は、単位時間内に発生する発熱量の平均値が制限値TH以下になるようにスケジューリング操作することを示しており、極短時間での制限値TH以上の発熱は平均化されるため問題とならなくなる。

【0027】次にこのような構成のマルチプロセッサシステムの処理動作例を図5のフローチャートを用いて説明する。図5を参照すると、まず、作成したソースリスト(プログラム)SLをコンパイラ9などでコンパイルして、単一のプロセッサで実行可能なオブジェクトコード(実行時のマシン語レベル)の命令に変換する(ステップS1)。このようにして、マシン語レベルでの命令(オブジェクトコード)が与えられると、命令割付手段1では、このコンパイル後の命令(オブジェクトコード)を各プロセッサP₁～P_nに割り付ける(ステップS2)。この処理は、具体的には、マシン語レベルに変換した命令(オブジェクトコード)をマルチプロセッサシステムに適応させるために各プロセッサ毎に命令を分配した場合のデータ依存関係を考慮して命令分割を行ない、各プロセッサへ割り付けを行なう。

【0028】次いで、依存関係解析手段7は、各プロセッサ毎に分割された命令中の他のプロセッサとのデータ依存関係、制御依存関係の解析を行なう(ステップS3)。しかる後、第1のスケジューリング手段2は、各プロセッサP₁～P_n毎に割り付けられた命令間のデータの同期をとり、命令のスケジューリングを行なう(ステップS4)。具体的には、データ依存関係、制御依存関係に基づき各プロセッサ毎に同期を取るためにそれぞれに命令実行順序の入れ替えや発熱量抑制機能を付加したウェイト命令の挿入等を行なう。

【0029】このようにして命令のスケジューリングがなされたとき、発熱量予測手段3では、命令ステップ毎にプロセッサ単体の発熱量を予測し(ステップS5)、総発熱量予測手段4では、各プロセッサP₁～P_nの命令ス

テップ毎の発熱量からチップ10全体の命令ステップ毎の発熱量を予測し、さらに、チップ全体の命令ステップ毎の発熱量を単位時間当たりに総和をとって総発熱量として予測する(ステップS6)。この段階で、第2のスケジューリング手段5は、総発熱量予測手段4による発熱量の予測結果に基づき、命令の再スケジューリングを行なう(ステップS7)。

【0030】具体的には、第2のスケジューリング手段5は、総発熱量予測手段4によって予測された総発熱量が制限値を越える場合には、この制限値を越えないように、第1のスケジューリング手段2によって選択された次の命令候補を他の命令の実行に置き換えるか、あるいは発熱を抑制するウェイト命令を挿入するように、各プロセッサP₁～P_nに対する命令の再スケジューリングを行ない、再スケジューリングした命令をチップ10上の各プロセッサP₁～P_nに組み込む。チップ10上の各プロセッサP₁～P_nは、再スケジューリングされた命令を実行し、この場合、上記再スケジューリングされた命令は、これを実行した場合にも各プロセッサP₁～P_nの総発熱量(チップ10全体の総発熱量)が制限値以下に抑制され、従って、処理を効率的に実行することができる。すなわち、チップ10内の発熱量を許容値以上にすることなく、命令の実行が可能となる。

【0031】上述の例では、発熱量予測手段3、総発熱量予測手段4で図2のような処理がなされるとしたが、図2のような処理のかわりに、図6のような処理を行なうようになっていても良い。すなわち、図6の処理例では、総発熱量予測手段4は、前記単位時間当たりの総発熱量の算出において、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算するとともに、単位時間経過した命令の発熱量を前記累積加算値から減算して、総発熱量を算出するようになっている。

【0032】具体的に、図6の処理例では、チップ内部のプロセッサP₁とプロセッサP₂が命令を実行しているとする場合、まず、図2の処理例と同様に、発熱量予測手段3は、プロセッサP₁の実行命令の命令コードC1により予測値テーブルからC1の発熱量H1を読み出し、また、プロセッサP₂の実行命令の命令コードC2により発熱量H2を読み出して、プロセッサ単体の命令ステップ単位の発熱量の予測値を算出する。

【0033】次いで、総発熱量予測手段4は、これら発熱量H1とH2の総和S1を求めることにより、その命令ステップでのチップ全体の発熱量S1(命令実行時間当たりの総発熱量)を得て、さらに、この発熱量S1を単位時間の間、累積加算し、この累積加算値により、単位時間当たりの総発熱量を得るが、この際、図6の処理例では、ある単位時間経過した命令の発熱量を上記累積加算値から減算し、これを総発熱量S2として算出する。すなわち、命令ステップ毎に新しい命令の発熱量を

加算し、対象エリアからはずれた過去の命令の発熱量は減算する。これにより、単位時間領域(対象エリア)を移動させることができる。換言すれば、図6の処理例では、単位時間領域(対象エリア)を命令ステップ毎(1命令実行時間毎)に区間移動する度に累積加算値S2を得るようになっている。

【0034】図7にはこの場合の発熱量算出の対象となる単位時間区間の例が示されている。この例では、命令ステップ毎に新しく選択された命令の発熱量を累積加算するとともに、ある単位時間経過した命令の発熱量を累積加算値から減算することにより、発熱量算出の対象となる単位時間領域を移動させることができる。

【0035】このように、総発熱量予測手段4は、図6の処理例では、命令ステップ毎に単位時間領域を移動させて、総発熱量S2を算出することができる。

【0036】なお、一般に、スケジューリングには動的スケジューリングと静的スケジューリングとがあるが、上述の例では、静的スケジューリングで考えている。すなわち、動的スケジューリングの場合には、実時間の計測処理となることから、熱管理用プロセッサや熱センサ等が必要になるが、静的スケジューリングの場合には熱管理用プロセッサや熱センサ等を必要としない。すなわち、プロセッサP₁~P_nを動かすプログラムデータの静的スケジューリングにおいては、プログラム実行を想定したチップ発熱の挙動について予めシミュレーションや実験等にて、命令ごとの発熱データを用意することにより、オフラインで熱管理処理を行なうことができる。

【0037】静的スケジューリングの処理流れとしては、前述したように、プログラムソースがコンパイルされた後、コンパイル(翻訳)された機械語命令コードは、命令割付手段1、第1のスケジューリング手段2、発熱量予測手段3、総発熱予測手段4、および第2のスケジューリング手段5を経て、熱管理処理された命令コードとなる。ただし、第2のスケジューリング手段5において、発熱を抑制するウェイト命令を挿入する場合には、ウェイト命令コードを実行するためのハードウェアが必要となる。この場合、発熱を抑制するウェイト命令コードが実行されたときの機能として、リソースに対するクロックラインの遮断機能や電圧低下機能を実現するハードウェア(スイッチングによる発熱を抑制する機能)が要求される。従って、この場合、第2のスケジューリング手段5については、これをチップ10上にハードウェアとしてもたせるのが良い。

【0038】また、図8は本発明に係るマルチプロセッサシステムの他の構成例を示す図である。図8を参照すると、このマルチプロセッサシステムの命令作成装置51は、基本的には、図1のマルチプロセッサシステムの命令作成装置11と同様の構成となっているが、図8の命令作成装置51では、第2のスケジューリング手段5から第1のスケジューリング手段2へのフィードバック

制御がなされるようになっている。

【0039】図9は図8のマルチプロセッサシステム(命令作成装置51)の処理動作例を示すフローチャートである。図9を参照すると、図8の命令作成装置51も、基本的には、図1の命令作成装置11の図5の処理動作と同様の処理動作を行なうが、図8の命令作成装置51では、ステップS7で発熱量を基にして再スケジューリングされた命令について、ステップS3で、再び、データ依存、制御依存関係の解析を行ない、ステップS4で同期のための命令の並び替えやウェイト命令の挿入などを行なう。

【0040】このようなフィードバックを所定回数繰り返して行なうことで、より効率的な命令のスケジューリングが可能となり、より効率の良い発熱制御が可能となる。

【0041】また、図10は図1あるいは図8の命令作成装置のハードウェア構成例を示す図である。図10を参照すると、この命令作成装置は、例えばワークステーションやパーソナルコンピュータ等で実現され、全体を制御するCPU21と、CPU21の制御プログラム等が記憶されているROM22と、CPU21のワークエリア等として使用されるRAM23と、ソースリストSLを入力する入力装置24と、作成した命令コードをチップ10上の各プロセッサに与える出力装置26とを有している。

【0042】ここで、CPU21は、図1あるいは図8のコンパイラ9、命令割付手段1、第1のスケジューリング手段2、発熱量予測手段3、総発熱予測手段4、および第2のスケジューリング手段5の機能を有している。

【0043】なお、CPU21におけるこのようなコンパイラ9、命令割付手段1、第1のスケジューリング手段2、発熱量予測手段3、総発熱予測手段4、および第2のスケジューリング手段5等としての機能は、例えばソフトウェアパッケージ(具体的には、CD-ROM等の情報記録媒体)の形で提供することができ、このため、図10の例では、情報記録媒体30がセットさせるとき、これを駆動する媒体駆動装置31が設けられている。

【0044】換言すれば、本発明の命令作成装置は、汎用の計算機システムにCD-ROM等の情報記録媒体に記録されたプログラムを読み込ませて、この汎用計算機システムのマイクロプロセッサに上述の処理を実行させる装置構成においても実施することが可能である。この場合、本発明の命令作成処理を実行するためのプログラム(すなわち、ハードウェアシステムで用いられるプログラム)は、媒体に記録された状態で提供される。プログラムなどが記録される情報記録媒体としては、CD-ROMに限られるものではなく、ROM、RAM、フレキシブルディスク、メモ리카ード等が用いられても良

い。媒体に記録されたプログラムは、ハードウェアシステムに組み込まれている記憶装置、例えばハードディスク装置にインストールされることにより、このプログラムを実行して、コンパイラ 9、命令割付手段 1、第 1 のスケジューリング手段 2、発熱量予測手段 3、総発熱量予測手段 4、および第 2 のスケジューリング手段 5 の機能を実現するマルチプロセッサシステムの構築に寄与する。

【0045】また、本発明のコンパイラ 9、命令割付手段 1、第 1 のスケジューリング手段 2、発熱量予測手段 3、総発熱量予測手段 4、および第 2 のスケジューリング手段 5 の機能を実現するためのプログラムは、媒体の形で提供されるのみならず、通信によって(例えばサーバによって)提供されるものであっても良い。

【0046】また、図 10 の例では、命令作成装置は、チップ 10 とは別体の装置として構成されているが、命令作成装置は、チップ 10 上に、プロセッサ $P_1 \sim P_n$ とともに実装されていても良い。すなわち、チップ 10 上に、プロセッサ $P_1 \sim P_n$ の他に、さらに、例えば、命令作成装置として機能する CPU と ROM などを実装し、命令自体を ROM から供給するか、あるいは、CPU で生成し、図 10 の出力装置 26 のような外部との入出力インタフェースを必要とせずに、ROM から供給される命令あるいは CPU で生成される命令をプロセッサ $P_1 \sim P_n$ に割り付けることもできる。換言すれば、プロセッサ $P_1 \sim P_n$ のみならず命令作成装置をも含めて 1 チップ上に実装したマルチプロセッサシステムとして構成することも可能であり、この場合には、1 チップ内で全ての制御がなされる。

【0047】

【発明の効果】以上に説明したように、請求項 1、請求項 2、請求項 4、請求項 5 記載の発明によれば、独立動作可能な複数のプロセッサがチップ上に実装されているマルチプロセッサシステムにおいて、コンパイル後の命令をチップ上の各プロセッサに割り付ける命令割付手段と、各プロセッサ毎に割り付けられた命令間のデータの同期をとるように命令のスケジューリングを行なう第 1 のスケジューリング手段と、命令ステップ毎にプロセッサ単体の発熱量を予測する発熱量予測手段と、各プロセッサの発熱量からチップ全体の発熱量を予測する総発熱量予測手段と、総発熱量予測手段によって予測された総発熱量からチップ上の各プロセッサ毎の命令の再スケジューリングを行ないチップ全体の発熱量を抑制する第 2 のスケジューリング手段とを有しており、発熱量を考慮した命令のスケジューリングを行ない、チップ全体の発熱量を抑えることにより、チップの発熱許容限界を越えることなく処理を行なうことができ、低消費電力化を図ることができる。

【0048】また、請求項 3、請求項 7 記載の発明によれば、前記第 2 のスケジューリング手段から前記第 1 の

スケジューリング手段へのフィードバックを行ない、前記第 2 のスケジューリング手段によって再スケジューリングされた命令を前記第 1 のスケジューリング手段によって再びスケジューリングする処理を、所定回数、繰り返し行なうので、より効率的な命令のスケジューリングが可能となり、より効率の良い発熱制御が可能となる。

【0049】また、請求項 6 記載の発明によれば、前記総発熱量予測手段は、前記単位時間当たりの総発熱量の算出において、命令ステップ単位でのチップ全体の発熱量の予測値を所定の命令ステップ毎に単位時間の間、累積加算するとともに、単位時間経過した命令の発熱量を前記累積加算値から減算して、総発熱量を算出するようになっているので、命令ステップ毎に単位時間区間を移動することができ、常に最新の実行ステップにおける発熱量の算出が可能となつて、きめ細かな低消費電力化を図ることができる。

【図面の簡単な説明】

【図 1】本発明に係るマルチプロセッサシステムの構成例を示す図である。

【図 2】発熱量予測手段、総発熱量予測手段の処理例の概要を示す図である。

【図 3】発熱量予測手段、総発熱量予測手段で図 2 のような処理がなされるとした場合の発熱量制御手段における処理の概要を説明するための図である。

【図 4】1 つのプロセッサ、例えば P_n に対する命令スケジュールの一例を示す図である。

【図 5】図 1 のマルチプロセッサシステムの処理動作例を説明するためのフローチャートである。

【図 6】発熱量予測手段、総発熱量予測手段の他の処理例の概要を示す図である。

【図 7】図 6 の処理例において、発熱量算出の対象となる単位時間区間の例を示す図である。

【図 8】本発明に係るマルチプロセッサシステムの他の構成例を示す図である。

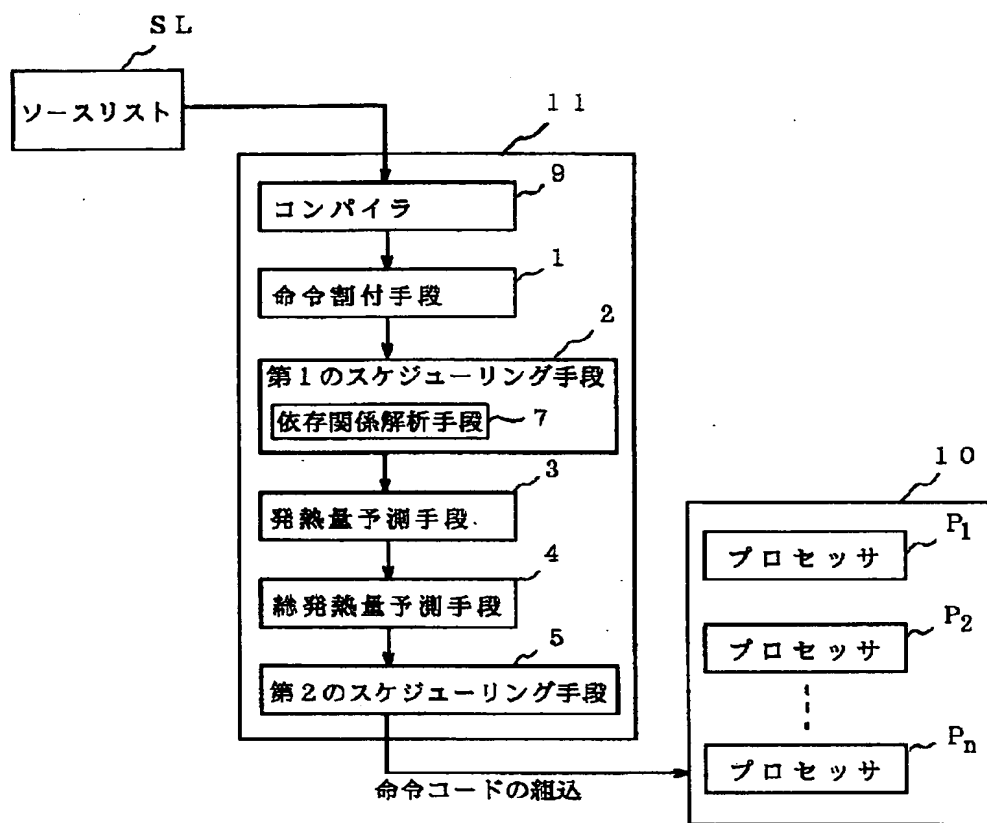
【図 9】図 8 のマルチプロセッサシステムの処理動作例を説明するためのフローチャートである。

【図 10】図 1 あるいは図 8 の命令作成装置のハードウェア構成例を示す図である。

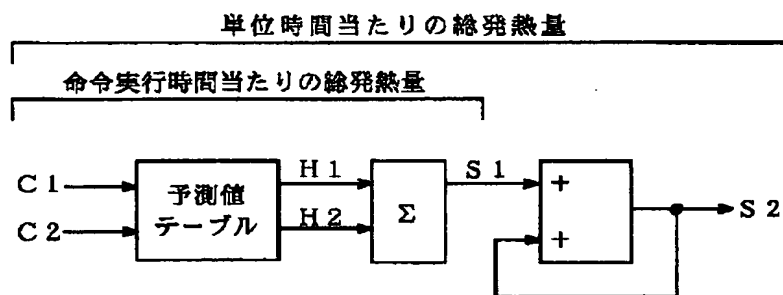
【符号の説明】

1	命令割付手段
2	第 1 のスケジューリング手段
3	発熱量予測手段
4	総発熱量予測手段
5	第 2 のスケジューリング手段
7	依存関係解析手段
9	コンパイラ
10	チップ
11	命令作成装置
$P_1 \sim P_n$	プロセッサ
SL	ソースリスト

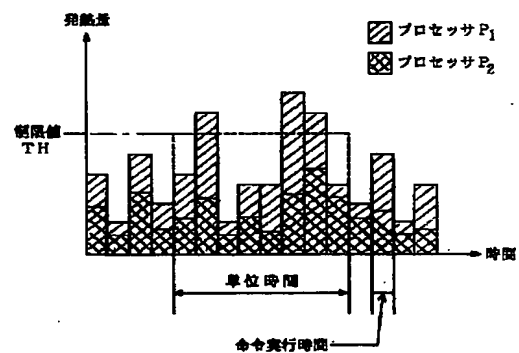
【図1】



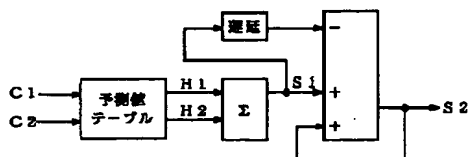
【図2】



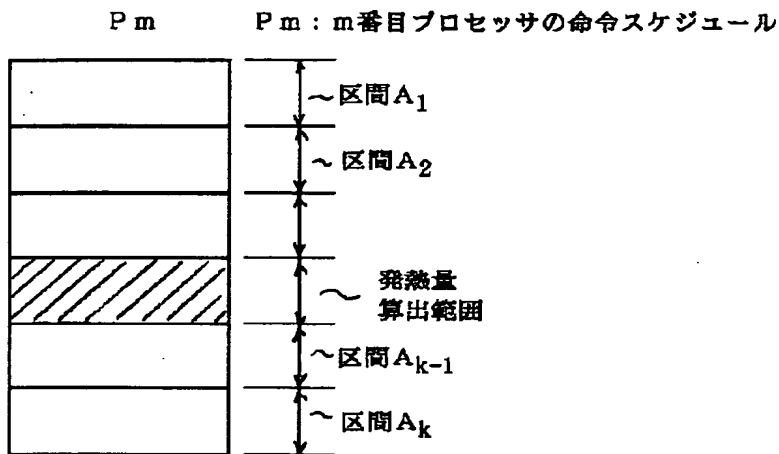
【図3】



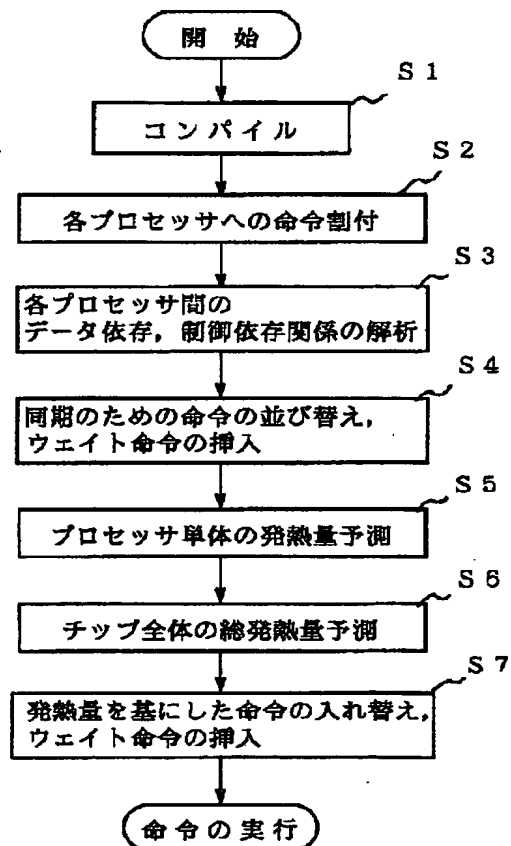
【図6】



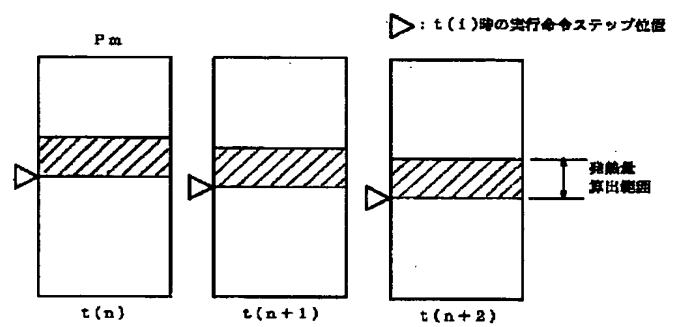
【図4】



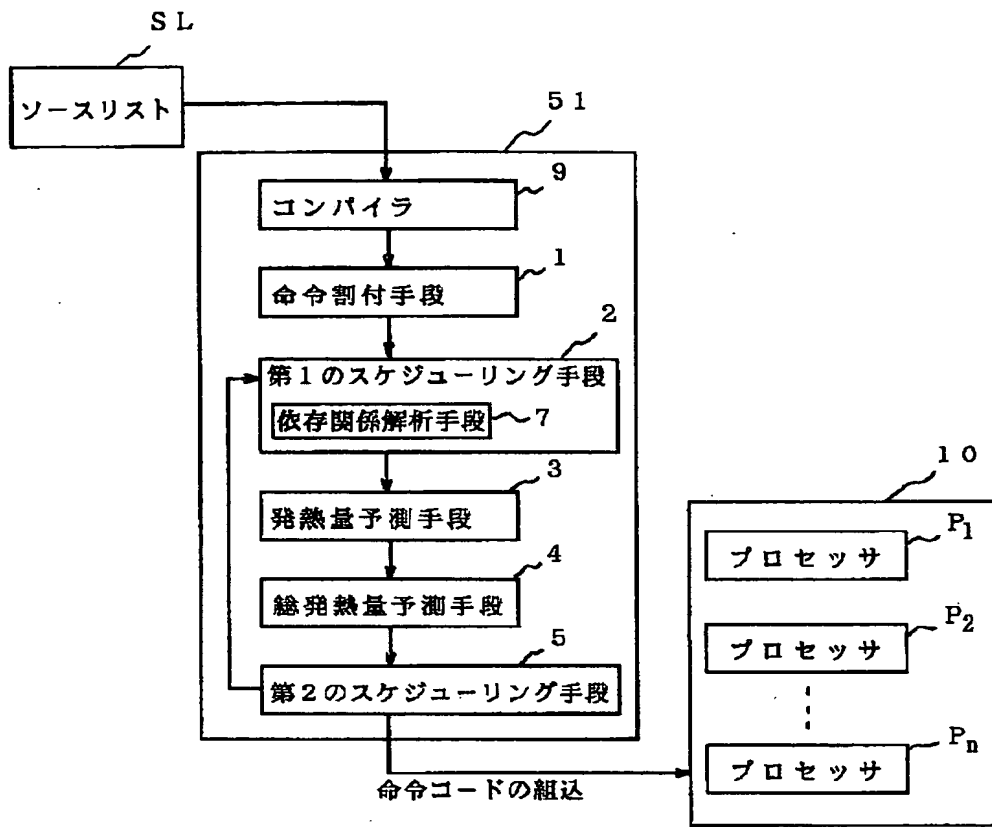
【図5】



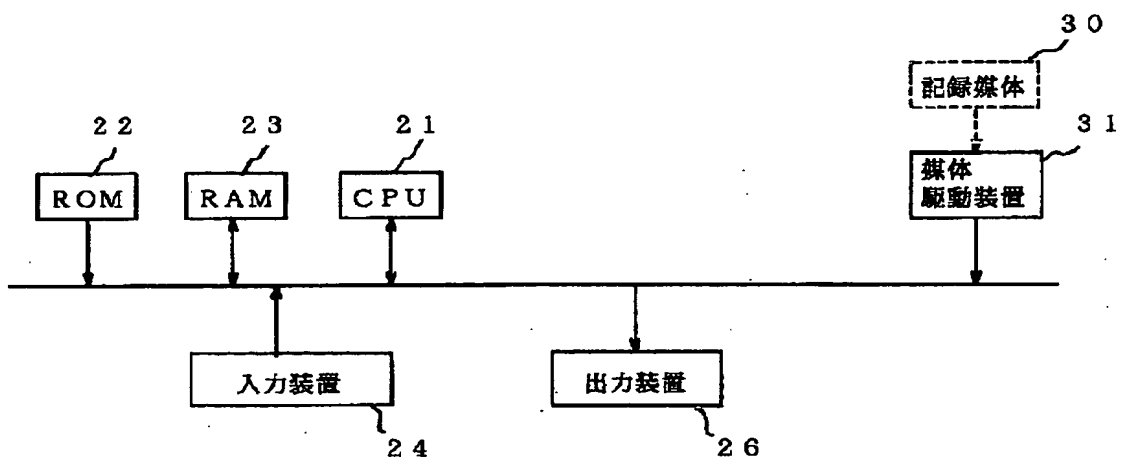
【図7】



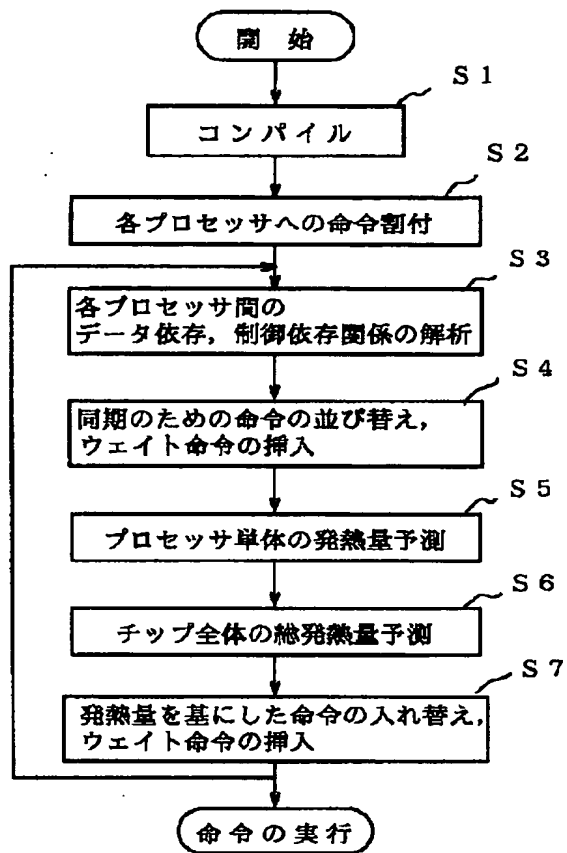
【図8】



【図10】



【図9】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.